AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Currently amended) A method for starting a group of enterprise servers belonging to cluster of enterprise servers, the method comprising:

receiving, in an enterprise server in a group of enterprise servers, the group belonging to a cluster of enterprise servers, the cluster having a central database accessible to the group of enterprise servers, a notification that binaries and/or configuration settings related to the cluster and stored within the central database have been modified;

comparing binaries and/or configuration settings stored within a local file system of each enterprise server with the modified binaries and/or configuration settings related to the cluster and stored within [[a]] the central database accessible to the group of enterprise servers to identify any binaries and/or configuration settings stored within the local file system which are out-of-date as compared to the binaries and/or configuration settings related to the cluster and stored within the central database;

if the binaries and/or configuration settings stored within the local file system are out-of-date as compared to the modified binaries and/or configuration settings related to the cluster and stored within the central database, then updating the modified binaries and/or configuration settings from the central database to the local file system prior to starting each enterprise server in the group of enterprise servers; and starting each enterprise server in the group of enterprise servers using the updated binaries and/or configuration settings stored within the local file system.

2. (Original) The method as in claim 1 further comprising:

generating a list of servers within the group to be started based on server layout information retrieved from the central database, the server layout information uniquely identifying each server in the group and/or parameters associated with each server in the group.

3. (Original) The method as in claim 2 wherein said layout information is defined by a configuration hierarchy stored within a hierarchical data object in the central database.

4. (Currently amended) The method as in claim 3 wherein the hierarchical data object comprises a global sub-hierarchy and a non-global sub-hierarchy, the global sub-hierarchy containing configuration data and binaries associated with all of the [[sever]] server nodes in the cluster and the non-global sub-hierarchy containing the layout information, configuration data and binaries associated with particular [[sever]] server nodes in the cluster.

5. (Original) The method as in claim 1 wherein the group of enterprise servers comprises an instance of enterprise servers.

6. (Original) The method as in claim 5 wherein the instance of enterprise servers comprises at least one dispatcher and two or more server

nodes.

7. (Original) The method as in claim 1 wherein the servers within the group comprise Java 2 Enterprise Edition ("J2EE") servers.

8. (Currently amended) A system for [[staring]] starting a group of enterprise servers belonging to a cluster of enterprise servers, the system comprising:

a central database for storing binaries and configuration settings associated with [[the]] a group of enterprise servers belonging to a cluster of enterprise servers;

receiving logic to receive a notification that the binaries and configuration settings associated with the group of enterprise servers belonging to the cluster of enterprise servers have been modified;

bootstrap logic to compare binaries and/or configuration settings stored within a local file system of each enterprise server with the modified binaries and/or configuration settings stored within the central database to identify any binaries and/or configuration settings stored within the local file system which are out-of-date,

wherein if the binaries and/or configuration settings stored within the local file system are out-of-date, then the bootstrap logic updates the modified binaries and/or configuration settings from the central database to the local file system prior to starting each enterprise server; and

startup and control logic to start each enterprise server in the group of enterprise servers using the updated binaries and/or configuration settings stored within the local file system.

9. (Currently amended) The system as in claim 8 wherein said

bootstrap logic comprises group bootstrap logic and node-specific bootstrap logic, the system further comprising:

group bootstrap logic to generate a list of servers within the group to be started based on server layout information retrieved from the central database, the server layout information uniquely identifying [[to]] the node-specific bootstrap logic, wherein the node-specific bootstrap logic comprises startup and control logic associated with each server in the group and/or parameters associated with each server in the group.

10. (Original) The system as in claim 9 wherein said layout information is defined by a configuration hierarchy stored within a hierarchical data object in the central database.

11. (Currently amended) The system as in claim 10 wherein the hierarchical data object comprises a global sub-hierarchy and a non-global sub-hierarchy, the global sub-hierarchy containing configuration data and binaries associated with all of the servers in the cluster and the non-global sub-hierarchy containing the layout information and configuration data and binaries associated with specified individual servers in the cluster.

12. (Original) The system as in claim 8 wherein the group of enterprise servers comprises an instance of enterprise servers.

13. (Original) The system as in claim 12 wherein the instance of enterprise servers comprises at least one dispatcher and two or more sever nodes.

14. (Original) The system as in claim 8 wherein the servers within

the group comprise Java 2 Enterprise Edition ("J2EE") servers.

15. (Currently amended) A system for preparing a group of servers for startup, the group of servers belonging to a cluster of servers, the system comprising:

a central database for storing binaries and configuration settings associated with a group of servers belonging to a cluster of servers, the central database accessible to the group of servers belonging to the cluster of server;

notification logic to receive a notification that the binaries and configuration settings associated with the group of servers belonging to the cluster of servers have been modified;

first bootstrap logic to retrieve layout data from [[a]] the central database, the layout data identifying the servers within the group and configuration parameters to be used for the servers within the group, the first bootstrap logic generating a data object identifying the servers and/or configuration parameters; and

second bootstrap logic to compare binaries and/or configuration settings stored within a local file system of each server with the modified binaries and/or configuration settings stored within [[a]] the central database ~~accessible to the group of servers~~ to identify any binaries and/or configuration settings stored within the local file system which are out-of-date as compared to the modified binaries and/or configuration settings stored within the central database,

wherein if the binaries and/or configuration settings stored within the local file system are out-of-date, then the second bootstrap logic updates the modified binaries and/or configuration settings from the central database to the local file system prior to starting each server within the group using the updated binaries and/or configuration settings stored within the local file system.

16. (Original) The system as in claim 15 further comprising: startup and control logic to start each server identified in the data object using the updated binaries and/or configuration settings.

17. (Original) The system as in claim 15 wherein said layout data is defined by a configuration hierarchy stored within a hierarchical data object in the central database.

18. (Currently amended) The system as in claim 17 wherein the hierarchical data object comprises a global sub-hierarchy and a non-global sub-hierarchy, the global sub-hierarchy containing configuration data and binaries associated with all of the servers in the cluster and the non-global sub-hierarchy containing the layout information and configuration data and binaries associated with specified individual servers in the cluster.

19. (Original) The system as in claim 15 wherein the group of servers comprises an instance of enterprise servers.

20. (Original) The system as in claim 19 wherein the instance of enterprise servers comprises at least one dispatcher and two or more sever nodes.

21. (Original) The system as in claim 15 wherein the servers within the group comprise Java 2 Enterprise Edition ("J2EE") servers.